Detecting and Analyzing Copy-Move Forgery of an image

[1]Kuwar Pallavi, [2]More Mayuri, [3]Kokani Nikita, [4]Sonawane Aishwarya

Research Scholar, Department of Computer Engineering
K. K. Wagh Institute of Engineering Education and Research

Abstract: **Copy-move forgery is the type of image tampering in which some part of image is copied and pasted on another part of same image. Previously, two methods were used to detect copy-move forgery namely point based method and block based method. But this method does not work well for geometric transformation that is scaling and rotation for homogenous area. In this paper, we propose a new approach in which use Delaunay triangulation method to form triangles and then we compare that triangles to get the results. For preprocessing, Otsu method is used. Our experimental results show that our proposed approach detects copied area in the forge images very accurately even though it includes geometric distortions. The proposed method improves the efficiency as compare to existing methods.**

Index terms: **Digital image forensics, Copy-move forgery, SIFT, SURF, Delaunay triangulation.**

1. Introduction

Digital Image Forensics deals with the problem of certifying the authenticity of a picture, or its origin. An image has always implied the truth of what it represents. The advent of digital pictures and relative ease of digital image processing makes today this authenticity uncertain. Digital images can be manipulated in such a perfect way that the forgery cannot be visually perceived by naked eye. Nowadays, in our society, we can come in contact with a lot of tampered images, in news report, business, law, military affairs, academic research. More particularly, tampered images could be used to distort the truth in news reports, to destroy someone's reputation and privacy, e.g. by changing a face of a person in a photo with someone else's face. The techniques used to verify the authenticity of an image can be further divided into two major groups: intrusive and non-intrusive. In intrusive (active) techniques, some sort of signature (watermark, extrinsic fingerprint) is embedded into a digital image, and authenticity is established by verifying if the retrieved signature matches the original one, or if it is corrupted. The use of active methods is limited, due to the inability of many digital cameras and video recorders to embed extrinsic fingerprints. Passive techniques use the intrinsic content of an image to detect if it has been tampered, without any superimposed information.

Detection of a special type of digital forgery – the copy-move attack in which a part of the image is copied and pasted somewhere else in the image with the intent to cover an important image feature. We investigate the problem of detecting the copy-move forgery and describe an efficient and reliable detection method. The method may successfully detect the forged part even when the copied area is enhanced to merge it with the background and when the forged image is saved in JPEG.

The proposed methods can be used in the future to find copies also in case of some other type of transformations, e.g. anisotropic deformations, as we divide the object into its atomic elements, and each of them can be separately analyzed. We plan also to develop some post processing techniques, to recover some missing matches, e.g. filling the holes between triangles, and to increase the recall of the methods.

Image Forensics techniques are used mainly when an image is presented as an official proof to influence the judgment. During last decade different techniques for validating the integrity of digital images have been developed. There are three main branches in Digital Image Forensics: First is Image Source Identification, that aims to identify which device was used to capture an image (model or exemplar of scanner, of digital camera); Second is Discrimination of Computer Generated Images, to detect if an image is natural or synthetic; and third is Image Forgery Detection, to discern if an image has been intentionally modified by human intervention. Image Forensics techniques are to understand what kind of tampering has been applied. Images can be doctored in several ways: photo-compositing, re-touching, enhancing are only some examples of typical image alterations. Although many tampering operations generate no visual artifacts in the image, they will nevertheless affect its inherent statistics.

The goal of copy move forgery is to replicate a part of an image, often to hide an object, by copy-pasting a set of pixels from an area to another area of the same picture, and it is often very difficult to detect with the naked eye.

1.1 PROPOSED APPROACH

In this approach we use set of connected triangles. We decided to apply it to our set of 2D images. We first extract points of interest from an image, using detectors named as SIFT and SURF. A Delaunay triangulation is built onto the extracted points. Image is therefore subdivided into triangles, which include pixels with very similar features. We decided to use the Delaunay triangulation as its atomic element typically does not include edges of the objects and its content may be considered as homogenous. Furthermore, to include also the outer parts of the image, where typically no interest points are extracted, we added uniformly arbitrary points onto the borders of the image. This solution does not influence the triangle mesh construction onto the extracted keypoints, but helps us to subdivide into triangles also the parts of the image that are near the vertical or the horizontal borders.

In this paper we present a method triangles matching by colors and angles to represent the object. In this method we analyze the features of triangles that is color and angles. Angles are taken in counterclockwise order. The input image is segmented into triangles based on extraction points of interest. To find copy moved regions, we first calculate colors and angles, and then we search for similar triangles into the image. First, triangles are sorted according to color vectors. The sorted list of triangles is then scanned and the features of each triangle are compared to the next triangles in the list, within a fixed window.

If both the Sum of the Absolute Deviation (SAD) of the color vectors and of the angles is below a threshold, the two triangles are considered similar. If j and k (k>j) are the indexes of the two triangles to be compared:

$$\sum_{i=1}^{3.n} |C_i^j - C_i^k| <= TH^c$$

$$(k - j) < w_s$$

$$\sum_{l=1}^{3} |a_i^j - a_l^k| <= TH^a$$

Where ws is the fixed window size, C is the color vector (made of 3*n values), are the angles in radians and two thresholds.
Our method is designed to determine copied objects in case of geometric transformation.
To reduce false positives, we compare two triangles j and k only if the ratio between their areas:

$$r_A = \frac{\min(A_j, A_k)}{\max(A_j, A_k)} >= 0.25$$

It deletes 20-25% of wrong matches. To further delete false matches, we compute set of matching centroids of triangles, and we apply RANSAC (RANdom SAmple Consensus) to select a set of inliers that are compatible with a homographic transformation. If less than 4 matches are found, RANSAC cannot be applied, and the match is considered “not reliable”.

1.2 Otsu’s Method

Otsu’s method is used to create better result. It is named after Nobuyuki Otsu is used to automatically perform clustering-based image thresholding or the reduction of a grey level image to a binary image. Otsu’s method is roughly one dimensional. The extension of the original method to multilevel thresholding is referred to as the Multi Otsu method.

In this method, we search for the threshold that maximizes the intra class variance, defined as a weighted sum of variance of the two classes:

Here, weights Probabilities of two classes separated by threshold t. = variances of these classes.
This method shows that maximizing intra-class variance is same as minimizing inter-class variance.

Where,

Class probability (t) is computed from the histogram as t:
(t)=
Class mean (t) is computed from:
(t)=
Where, x (i) = value at canter of histogram bin.
and can be computed iteratively. This idea yields an effective algorithm. Otsu method produces a threshold on the 0:1 scale.

Fig: pre-processing Otsu's method

Fig: post-processing Otsu's method

1.3 Algorithms

2.1 SIFT Algorithm

There are mainly 4 steps of SIFT algorithm.

1) Scale-space Extreme Detection

For small corners of image it requires small window but for larger corners it required larger window. So for this purpose scale space filtering is used. In it Laplacian of Gaussian is found for image with various values. LoG acts as blob detector which detects blob in various sizes due to change in that is act like scaling parameter. LoG is costly so this algorithm uses difference of Gaussian is obtained as the difference of Gaussian blurring of an image. This process is done for different octaves of image in Gaussian pyramid. Once DoG is found images are searched for local maxima over scale and space. If it local extreme it is a potential keypoint is best represented in that scale.

2) Keypoint Localization

Once potential keypoints locations are found refined it for accurate results. If intensity of extrema is less than a threshold value it is rejected. DoG has higher response for edges, so we have to remove edges. It use 2*2 matrix to compute principal curvature. If ratio of Eigen values is greater than threshold that keypoint is discarded. So it eliminates any low contrast keypoints and edge keypoints then the remaining are strong interest points.

3) Orientation Assignment

To achieve invariance to image rotation, orientation is assign to each keypoint. A neighborhood taken around keypoint location and then calculate magnitude and direction. An orientation histogram with 36 bins covering 360 degree is created.

4) Keypoint Descriptor

A 16x16 neighborhood around the keypoint is taken. It also subdivided into 4x4 sizes. For each sub-block, 8 bin orientation histogram is created. So total 128 values are available. It term as vector called keypoint descriptor.

5) Keypoint Matching

Keypoints between two images are matched by identifying their nearest neighbours. In some cases, second closest match is very near to first because of noise. In that case, ratio of closest distance to second closest distance is taken and if it is greater than threshold they are rejected.

This algorithm eliminates 90% of false matches and 5% of correct matches.

2.2 SURF Algorithm:

It has three main parts.

1. Interest point detection.
2. Local neighborhood description.
3. Matching

- **Interest point detection-**

The SIFT approach is user cascaded filters to detect scale invariant characteristic point, where the difference of Gaussian (DoG) is calculated on rescaled images progressively.

In SURF, square shaped filters are used as an approximation of Gaussian smoothing. Filtering the image with a square is much faster if the integral image is used, which is

S(x, y)

The sum of the original image within a rectangle can be evaluated quickly using the integral image requiring four evaluations at the corners of rectangle. SURF uses a blob detector based on the hessian matrix to find points of interest. The determinant of the Hessian Matrix is used as a measure of local change around the point of points is chosen where this determinant is maximal. In contrast to the Hassian-Laplacian detector, SURF also uses the determinant of the Hessian for selecting the scale, as if is done by lindeberg. Given a point p=(x,y) in an image I, the Hessian matrix H(p,sigma) at point and scale sigma is,

H(p, sigma)= (Lxx(p, sigma) Lxy(p, sigma)
Lxy(p, sigma) Lyy(p, sigma))

Where Lxx (p, sigma) are the second order derivatives of the grayscale image.

Scale-space representation and location of point of interest –

The interest points can be found in different scale partly because the search for correspondences often require comparison images where they are seen at different scales. In other feature detection algorithm, the scale space is usually realized as an image pyramid.

Images are repeatedly smoothed with a Gaussian filter. Then they are subsamples to get the next higher level of the pyramid. There several floors or stairs with various measures of the masks are calculated.

Sigma(approx)= Current filter size*(Base filter scale/Base filter size)

The space is divided into a number of octaves where refers to a series of response maps of covering a doubling of scale. In SURF, the lowest level of the scale. Space is obtained from the output of the 9*9 filters.

Hence, unlike previous methods, scale spaces in SURF are implemented by applying box filters of different sizes. Therefore, the scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size. The output of the above 9*9 filter is considered as the initial scale layer, to which we will refer as scale s=1.2.

The following layers are obtained by filtering the image with gradually bigger masks taking into account the discrete nature of integral images and the specific structure of filters.

- **Local neighborhood descriptor-**

The goal of a descriptor is to provide a unique and robust description of an image feature. E.g., by describing the intensity distribution of the pixel within the neighborhood of the point of interest. Most descriptor are computed thus in a local manner, hence a description is obtained for every point of interest identified previously.

The dimensionality of the descriptor has direct impact of both its computational complexity and point matching robustness/accuracy. A short descriptor may be more robust against appearance variations but may not offer sufficient discrimination and thus give too many false positive.

- **Matching**

By comparing the descriptors obtained from different image, matching pairs can be found.

2.3 **Otsu Algorithm:**

- Compute histogram & probabilities of each intensity level.
- Set up initial &
- Step through all possible thresholds t = 1 . . . . . maximum intensity.
    - Update &

- Compute
- Desired threshold corresponds to the maximum.
- You can compute two maxima (& two corresponding thresholds).
  - Is the greater max & is the greater or equal maximum.
- Desired threshold =

2.4 Canny Edge Detector Algorithm:

Canny Edge Detector Algorithm is an edge detection operator that uses a multi stage algorithm to detect a wide range of edges in images.

Algorithm:

- Apply Gaussian filter to smooth the image in order remove the noise.
  Equation for Gaussian filter kernel size (2k+1)*(2k+1)
- Find the intensity gradients of the image.
  Edge detector operator returns a value for the first derivative in horizontal direction (G_x) and vertical direction (G_y). From this edge gradient and direction can be determined.
- Apply non-maximum suppression of get rid of spurious response to edge detection.
  Is an edge thinning technique. It can helps to suppress all the gradient values to 0 except the local maximal ,which indicate location with the sharpest change of intensity value.
  If rounded gradient angle is 0^0 then edge in the north-south direction.
  If rounded gradient angle is 45^0 then edge in the northwest-southeast direction.
  If rounded gradient angle is 90^0 then edge in the east-west direction.
  If rounded gradient angle is 135^0 then edge in the northeast-southwest direction.

4. Apply double threshold to determine potential edges.
   Two threshold values are set to clarify the different types of edge pixels.
   One is called High threshold value and other is called low threshold value.
   If the edge pixels gradient value is higher than the high threshold value, they are marked as strong edge pixel.
   If the edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value , they are marked as weak edge pixels.
   If the pixel's value is smaller than low threshold value, they will be suppressed.
5. Track edge by hysteresis:
   Finalize the detection of edges by suppressing all other edges that are weak and not connected to strong edges.
   To track the edge connection, binary large object analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels.
   As long as there is one strong edge pixel is involved in the BLOB, that weak edge point can be identified as one that should b preserved.

2.5 Gray scale Conversion:

Any digital camera is capable of taking black and white photographs. Black and white photography is actually consists of many shades of gray. Color space is a way to visualize shapes or object that represents all available colors. Different ways of representing color lead to different color shapes. For example RGB model is of cube shape, HSL model is of cylindrical or cone shaped. There are three color channels- red channel, green channel, and blue channel. Different color models have different channels.

All gray scale algorithms utilize the same basic three-step process:-

- Get the red, green and blue values of pixel.
- Use fancy math to turn those numbers into a single gray value.
- Replace the original red, green and blue values with the new gray values.

For fancy math, we make use of following formula,

Gray = (Red + Green + Blue)/3

- Code / Algorithm:

```
For each pixel in image {
Red = pixel.Red
Green = pixel.Green
Blue = pixel.Blue
Gray = (Red + Green + Blue)/3
pixel.Red = Gray
pixel.Green = Gray
pixel.Blue = Gray
}
```

Method 1 for gray scale conversion
Averaging (quick and dirty)

This is the most common method used for grayscale conversion. It is fast and simple method and it works like:

Gray = (Red + Green + Blue)/3
It's simplicity makes it easy to implement and optimize. It does a poor job of representing shades of gray relative.

Method 2 for gray scale conversion
Correcting for the human eye

Cone density of human eye is not uniform across colors. Human perceive green more strongly than red and red more strongly than blue. Much natural world appears in shade of green, so human have evolved greater sensitivity to green light(oversimplified but accurate). Because humans cannot perceive all colors equally, the "Average method" of grayscale is inaccurate.
Gray = (Red * 0.3 + Green * 0.59 + Blue * 0.11)
This formula requires a bit of extra computation, but it results in a more dynamic grayscale image.
It's worth noting that there is disagreement on the best formula for this type of grayscale conversion. Here, we have chosen to go with the original ITU-R recommendation (BT.709, specifically) which is the historical precedent. This formula, sometimes called Luma, looks like this:
Gray = (Red * 0.2126 + Green * 0.7152 + Blue * 0.0722)
Some modern digital image and video formats use a different recommendation (BT.601), which calls for slightly different coefficients:
Gray = (Red * 0.299 + Green * 0.587 + Blue * 0.114)
A full discussion of which formula is "better" is beyond the scope of this article. For 99% of programmers, the difference between these two formulas is irrelevant. Both are perceptually preferable to the "average method" discussed.

Method 3 for gray scale conversion

De-saturation

Mostly RGB model is used, but human eye cannot visualize RGB that's why HSL is sometimes used. HSL stands for hue, saturation and lightness. Here, hue is name of the color (Red, Green, Orange, Yellow). Saturation describes how vivid a color is; a very vivid color has full saturation, while gray has no saturation. Lightness describes the brightness of a color; white has full lightness, while black has zero lightness.

De-saturating works by converting an RGB triplet to an HSL triplet, then forcing the saturation to 0. Basically it takes color and converts it to its least-saturated variant. A pixel can be de-saturated by finding the midpoint between maximum of (Red, Green, Blue) and the minimum of (Red, Green, Blue).
Gray = (max (Red, Green, Blue) + min (Red, Green, Blue)) / 2
In terms of the RGB color space, de-saturation forces each pixel to a point along the neutral axis running from (0, 0, 0) to (255, 255, 255).
De-saturation results in a flatter, softer grayscale image. If you compare this de-saturated sample to the human-eye-corrected sample (Method #2), you should notice a difference in the contrast of the image. Method #2 seems more like an Ansel Adams photograph, while de-saturation looks like the kind of grayscale photo you might take with a cheap point-and-shoot camera. Of the three methods discussed thus far, de-saturation results in the flattest (least contrast) and darkest overall image.

Method 4 for gray scale conversion

Decomposition

To decompose an image, we force each pixel to the highest (maximum) or lowest (minimum) of its red, green and blue values. Decomposition is done on per-pixel basis- if we are performing a maximum decompose and pixel1 is RGB (255, 0, 0) while pixel2 is RGB(0, 0, 64), we will set pixel1 to 255 and pixel2 to 64. Decomposition only cares about which color value is highest or lowest – not which channel it comes from.
Maximum decomposition Gray = max (Red, Green, Blue)
Minimum decomposition Gray = min (Red, Green, Blue)
Maximum decomposition provides a brighter grayscale image, while a minimum decomposition provides a darker one.

3. EXPERIMENTAL RESULTS

Dataset:
It is made of medium sized images (almost all 1000× 700 or 700 × 1000) and it is subdivided into several datasets. The first dataset D0 is made of 50 not compressed images with simply translated copies. For the other two groups of images (D1, D2) we selected 20 not compressed images, representing single object, simple background, rather than complex scenes, as we are interested in studying primarily the robustness against some specific attacks. The dataset D1 has been created by copy-pasting objects after rotation, D2 applying scaling to the copies. Each dataset has been further subdivided into subsets. The first subset D1.1 has been created applying to the copies 11 different types of rotation around the angle zero in the range of [−25°, 25°] with step 5°. The second subset D1.2 is created by rotating the copies by 12 different angles in the range of [0°, 360°] with a step of 30°. The third subset D1.3 is built by rotating the copies by 11 different angles in the range of [−5°, 5°] with a step of 1°. D1 is then composed by 680 images. The subset D2.1 is obtained by scaling the copies by 8 different scaling factors in the range of [0.25, 2] with step 0.25. In D2.2 copies are scaled by 11 scaling factors in the range of [0.75, 1.25] with step 0.05. D2 is then

composed of 380 images (with some intersections). We furthermore tested our approaches onto the 50 original images of the dataset D0 without tampering (subset D3), to study the ability to discriminate between tampered and not tampered images.

4. Summary and conclusion

As Otsu's method is widely used as a pre-processing step to segment images for further processing, it is important to achieve a high accuracy. However, since Otsu's threshold is biased towards the class with a large variance, it tends to miss weak objects or fine details in images. Without a robust segmentation results, more sophisticated processing such as tracking and feature analysis become highly challenging. Otsu's method is optimal to separate a bi-modal histogram into two classes where the probability distribution functions (PDFs) of the two classes have an approximately "tall and thin" shape.

Block matching method is good in case of pure translation and gives information about copied pixels. And they work in homogeneous areas. They have better performance in case of complex scenes. Point based method is good in case of geometrical transformation. It gives information about single point that are part of copy-pasted area but not about pixels inside copied areas. It cannot be used to detect copy-pasted areas unless proper post processing technique is used.

In our proposed approach, we analyze structure of objects in tampered images represented as connected triangles. This approach is used for copy-move recognition and detection as they are able to find presence of copy-move areas and to expose parts of images.

References

[1] X. Pan and S. Lyu, "Region duplication detection using image feature matching," IEEE Trans. Inf. Forensics Security, Dec. 2010.
[2] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy–move attack detection and transformation recovery," IEEE Trans. Inf. Forensics Security, Sep. 2011.
[3] C. Harris and M. Stephens, "A combined corner and edge detector," in Proc. 4th Alvey Vis. Conf.
[4] Liu et al., "Learning to detect a salient object," IEEE Trans. Pattern Anal. Mach. Intell., Feb. 2011.
[5] R. Dyer, H. Zhang, and T. Möller, "A survey of Delaunay structures for surface representation," GrUVi Lab, Burnaby, BC, Canada, Schooln Comput. Sci., Tech. Rep. TR 2009-01, 2009.